Logical N-AND Gate on a Molecular Turing Machine

Victor Hernandez-Urbina[1]

Institute of Perception, Action and Behavior, University of Edinburgh.
j.v.hernandez-urbina@ed.ac.uk

Abstract. In Boolean algebra, it is known that the logical function that corresponds to the negation of the conjunction –NAND– is universal in the sense that any other logical function can be built based on it. This property makes it essential to modern digital electronics and computer processor design. Here, we design a molecular Turing machine that computes the NAND function over binary strings of arbitrary length. For this purpose, we will perform a mathematical abstraction of the kind of operations that can be done over a double-stranded DNA molecule, as well as presenting a molecular encoding of the input symbols for such a machine.

Keywords: molecular computing, DNA computing, Turing machine, Boolean functions, restriction enzymes.

1 Introduction

Moore's Law is the empirical observation that the number of transistors on integrated circuits doubles approximately every two years. However, it is known that the continuous miniaturization of electronic devices has undesirable consequences: fitting more and more chips in an area that is becoming smaller according to the standards of the market results in an uncontrollable increase of heat when the device begins its operation [**?**]. Hardware manufacturers are reaching the physical limit when attempting to develop more powerful devices in the the least space possible. Moreover, the holy grail of digital storage is being able to store a bit per atom, which implies massive storage in a very small volume. However, theory says that in order to achieve such an endeavour, one would have to pay the quantum price for dealing with subatomic particles.

An alternative to electronic and quantum computing is *molecular computing*, which also attempts to reduce the scale of computation and at the same time increase its power. Molecular computing uses as hardware (rather *wetware*) organic molecules such as ribonucleic acid (RNA) and deoxyribonucleic acid (DNA). In this work, we focus on the latter, which is known elsewhere as DNA computing.

There are two main reasons that make DNA an attractive computing medium, one is *parallelism* and the other, *complementarity*. The density of information encoded in a DNA strand, plus its relatively ease to generate copies of it (via techniques such as PCR [**?**]), offers the possibility of realising parallel computing

massively. For example, Leonard Adleman, one of the pioneers of this computing paradigm, presented in [**?**] a solution to the *Hamiltonian Path Problem* by probing in parallel different solutions to this NP-complete problem. On the other hand, complementarity refers to the fact that a strand of DNA is composed by two single strands of nucleotides that *bind naturally* through their bases. There are four of such bases: guanine (G), adenine (A), thymine (T), and cytosine (C). Given their chemical composition, nucleotides with base G bind naturally to nucleotides with base C, this fact makes them complementary bases, whereas nucleotides with base A bind to those with base T. As mentioned before, this complementarity is given free by nature and it can be exploited to model and encode a particular task.

Representing a double strand of DNA as two single strands, one above the other taking into account the complementarity of their bases, is a big simplification when actually both strands are coupled forming the well-known double helix. Our representation of such strings in terms of a strand that extends itself horizontally will help us through our exposition in the following pages. However, we should also mention that in this work we adopt the convention of expressing the direction of a single string of DNA as going from the $5'-$end to the $3'-$end, which is given by the nature of its chemical components and that is used extensively in biochemistry [**?**].

DNA is one of the main computing media in molecular computation, and it is crucial in the processes inside the living cells of organisms. Its two main functions are the codification for the creation of new proteins, and self-replication, such that an exact copy of itself is inherited to cellular offspring. As a computing medium it has been used to solve the Hamiltonian Path Problem and other combinatorial problems [**?**], as well as being used as a mechanism to detect the presence of molecules that signal the emergence of a chronic disease [**?**][**?**]. In [**?**] we present the design of an enzyme-free molecular machine that detects the emergence of hepatic fibrosis, and upon detection releases a molecule to interfere with the over expressed gene.

In this paper, our motivation comes from the work of Vineet Gupta et al. [**?**] where the authors present a technique to simulate logic and arithmetic functions through DNA molecules. However, the issue that arises in such work is that the method suggested lacks proper automation, that is, depends greatly in the presence of human operation, which contradicts the essence of computation, namely, the automation of processes that would imply a big effort, cost, and time in a situation in which those resources are limited. Our suggestion, thus, is the design of a DNA machine, which mimics the behaviour a particular logical function autonomously, that is, minimising the intervention of an external operator. The logic function that we will simulate molecularly is the inverted AND gate, also known as NAND gate, which is functionally complete. This fact implies that any other logic function can be expressed as a combination of gates of this type. Hence, its importance in the design of digital electronics, such as computer processors.

We make use of a set of operations that can be performed on a DNA strand. This set can be regarded as a gift from biochemistry to this novel computing paradigm. The kind of operations that can be performed on a DNA strand are: duplication, concatenation, cleavage, extension, length measurement, among others. For the type of problem that we address here, we need only a small set of operations, which we briefly describe below. For a more exhaustive description we refer the reader to [**?**].

- *Measuring a DNA strand's length.* The length of a double strand of DNA is given by the number of base pairs (a nucleotide and its complement) and is denoted by the unit *bps*. To measure this length a technique known as gel electrophoresis is employed.
- *Concatenation and separation of a DNA strand.* DNA strands are joined by a covalent bond horizontally (that is, when nucleotides are adjacent in a single strand), and by a hydrogen bond vertically (that is, when bases are complementary). The latter is weaker than the former, so it is possible to disjoin two complementary strings of DNA leaving the nucleotides in each single string intact just by increasing temperature. If the solution is cooled down again, the single strings are susceptible of joining again. This process must be performed slowly so strings have a chance to recognise their complementary bases. Then, a type of enzyme known as ligase is used, which joins covalently two consecutive nucleotides [**?**].
- *Cutting a DNA strand.* A double-stranded DNA molecule can be cleaved by the action of enzymes. Their action can be performed at the ends of a molecule or within it. The latter type of enzymes is known as endonucleases, or restriction enzymes. These are found in bacteria and they provide their hosts with a defense mechanism against viruses. These enzymes acquire their name from the organism from which they were obtained. For example, enzyme EcoRI receives its name because it was the first enzyme to be identified (hence, *I*) in the bacteria *E. coli* strain R. Restriction enzymes work by identifying a particular DNA string and then cleaving the molecule either at this recognised site or a few nucleotides away from it. In this work, we use this latter type of enzymes, which are known as restriction enzymes type IIS [**?**].

With these ideas in mind, we present our model in the following pages. The structure of this paper is as follows, in the next section we present the mathematical version of the Turing machine that computes the logical NAND function. In section 2.1, we present the molecular version of this Turing machine, and we present the main result of our work, which is in the shape of a theorem summarising the mathematical abstraction of the elements comprising the molecular machine. Finally, in section 3 we discuss some of the implications of our model along with some conclusions.

2 Model

Broadly speaking, a Turing Machine (TM) is a mathematical object that mimics the action of a machine consisting of a *head* that reads and writes on a tape obeying a fixed set of rules. Although simple in its design, a Turing machine is a powerful computing device as it can be adapted to mimic the logic of any computer algorithm.

In theory, the tape of the machine has infinite length and its head moves one square at a time along it according to a program specified as a set of rules. Formally, a TM is defined as follows:

Definition 1. *A Turing Machine is the* $7-$*tuple* $(S, \Sigma, \beta, \Sigma^*, T, s_0, A)$*, which consists of:*

1. *A finite set of states* S*.*
2. *A finite set of symbols (or alphabet)* Σ*.*
3. *A symbol* $\beta \in \Sigma$ *representing a blank character.*
4. *A finite set* $\Sigma^* \subset \Sigma \setminus \{\beta\}$ *of input symbols.*
5. *A transition function* $T : S \times \Sigma \to S \times \Sigma \times \{L, R\}$*, where* L *(*R*) denotes a single left (right) movement of the head on the tape.*
6. *An initial state* $s_0 \in S$*.*
7. *A finite set of accepting states* $A \subset S$*, for which the machine halts and the computation is over.*

As mentioned before, the logic function that we implement on a TM is an inverted AND gate, whose truth table is shown in table 1. Our TM reads two binary strings of length n from its tape, and then computes the NAND function entry by entry following the rules specified by its truth table. As a result, the machine writes a binary string of length n on its tape with the result of the computation.

Table 1. NAND truth table

p	q	p NAND q
0	0	1
0	1	1
1	0	1
1	1	0

The first convention that we will adopt concerns the way in which we write the input strings on the machine's tape. Let $A = (a_1, a_2, \ldots, a_n)$ and $B = (b_1, b_2, \ldots, b_n)$ be two binary strings of length n. We write the strings A and B intercalating each element of their entries on the tape. That is:

$\cdots$	a_1	b_1	a_2	b_2	$\cdots$	a_n	b_n	$\cdots$

With this in mind, we define our TM in the following way. Our alphabet is the set $\Sigma = \{0, 1, \varepsilon, \beta\}$, where 0 and 1 are the characters comprising our binary input and output strings, ε is a symbol that will serve as an error detection mechanism at the end of the computation, and β is our blank character. Thus, $\Sigma^* = \{0, 1\}$. We consider the set of states $S = \{S_0, S_1, S_2, HALT\}$, where $HALT \in A$ represents the accepting state which stops the computation, and S_0 is the initial state of the machine. Finally, the transition rules of our machine, and which ultimately mimic the behaviour of a NAND gate, are the following:

$$\langle S_0, 0\rangle \mapsto \langle S_1, \beta, \Rightarrow\rangle \tag{1}$$

$$\langle S_0, 1\rangle \mapsto \langle S_2, \beta, \Rightarrow\rangle \tag{2}$$

$$\langle S_0, \beta\rangle \mapsto \langle HALT\rangle \tag{3}$$

$$\langle S_1, 0\rangle \mapsto \langle S_0, 1, \Rightarrow\rangle \tag{4}$$

$$\langle S_1, 1\rangle \mapsto \langle S_0, 1, \Rightarrow\rangle \tag{5}$$

$$\langle S_1, \beta\rangle \mapsto \langle S_0, \epsilon, \Rightarrow\rangle \tag{6}$$

$$\langle S_2, 0\rangle \mapsto \langle S_0, 1, \Rightarrow\rangle \tag{7}$$

$$\langle S_2, 1\rangle \mapsto \langle S_0, 0, \Rightarrow\rangle \tag{8}$$

$$\langle S_2, \beta\rangle \mapsto \langle S_0, \varepsilon, \Rightarrow\rangle \tag{9}$$

Each of the transitions are interpreted in the following way:

$$\langle current\_state, character\_being\_read\rangle \mapsto \langle new\_state, character\_to\_write, move\_head\_to\rangle$$

In the following section we will implement all the elements of this TM into DNA molecules and we will provide a mathematical proof of its operation.

2.1 Molecular Turing Machine

In this section we present the molecular version of the TM defined previously. We perform a molecular encoding of all the elements of the TM recently described, which means that we will represent the alphabet, the transitions and states as DNA strands. In summary, a circular double-stranded DNA molecule of DNA will act as the tape of the TM, in which a substring will simulate the action of the head. Using ideas from [**?**], we use restriction enzymes to design the hardware of our machine. As mentioned previously, we use type IIS restriction enzymes to cut DNA strands at a defined distance from their non-palindromic asymmetric recognition sites [**?**], which in other words means that the recognition and the cleavage sites are away from each other.

In table 2 we describe the five restriction enzymes that we employ, their recognition site, the distance between it and the place the cut is performed,

and the direction in which they perform such cut. The nucleotide N_i denotes an arbitrary base. We must point out that these enzymes cut from right to left, or vice versa, whenever they find their recognition sites in the top string (in direction $5'$ to $3'$) or its mirrored version in the bottom string (in direction $3'$ to $5'$). However, we only specify the direction of cleavage in the way we require for the purposes of our machine. After cleavage these enzymes leave a sticky-end in the strings that have been cut. This sticky-end is an overhang of nucleotides in which another molecule can be joined. We group these five enzymes in a mathematical set that we name Φ.

Table 2. Restriction Enzymes

Name	Direction	Recognition Site	Restriction Site
FokI	$\rightarrow$	$5'-$GGATG$-3'$ $3'-$CCTAC$-5'$	$5' - \ldots N_9$ $3' - \ldots N_9N_{10}N_{11}N_{12}N_{13}$
BsrDI	$\rightarrow$	$5'-$GCAATG$-3'$ $3'-$CGTTAC$-5'$	$5' - N_1N_2$ $3' - \ldots$
BpmI	$\rightarrow$	$5'-$CTGGAG$-3'$ $3'-$GACCTC$-5'$	$5' - \ldots N_{14}N_{15}N_{16}$ $3' - \ldots N_{14}$
BpmI	$\leftarrow$	$5'-$CTCCAG$-3'$ $3'-$GAGGTC$-5'$	$N_{14} \ldots - 3'$ $N_{16}N_{15}N_{14} \ldots - 5'$
BserI	$\leftarrow$	$5'-$CTCCTC$-3'$ $3'-$GAGGAG$-5'$	$N_8 \ldots - 3'$ $N_{10}N_9N_8 \ldots - 5'$
BbvI	$\leftarrow$	$5'-$GCTGC$-3'$ $3'-$CGACG$-5'$	$N_{12}N_{11}N_{10}N_9N_8 \ldots - 3'$ $N_8 \ldots - 5'$

To simplify our exposition we adopt the following conventions regarding the representation of DNA strands, in particular, the representation of the recognition sites and cleavage direction of the restriction enzymes.

Notation 1. *A DNA strand (that is not linked by its extremes) will be represented as a string of characters enclosed by square brackets.*

$$[\qquad]$$

Whereas, a DNA strand that is linked by its extremes will be represented as a string of characters enclosed by simple parentheses.

$$(\qquad)$$

Notation 2. *The string* $\left[\overrightarrow{RS}\right]$ *denotes the recognition site of restriction enzyme* $RS \in \Phi$ *whose cleavage direction is from left to right. Analogously, the string* $\left[\overleftarrow{RS}\right]$ *denotes the same enzyme, but this time its cleavage is from right to left.*

We will use a word size of 6 bps to encode the symbols in the set Σ, plus a suffix of 4 bps, which serves as character delimiter in the molecular tape. We do not give an explicit declaration of the bases comprising the strands from the alphabet; as long as they do not contain any of the recognition sites used in the set Φ, the choice of their bases is arbitrary. Thus, our symbols, plus the suffix, look like:

$$0 = \left[\frac{A_1A_2A_3A_4A_5A_6}{\widetilde{A_1}\widetilde{A_2}\widetilde{A_3}\widetilde{A_4}\widetilde{A_5}\widetilde{A_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right] \tag{10}$$

$$1 = \left[\frac{B_1B_2B_3B_4B_5B_6}{\widetilde{B_1}\widetilde{B_2}\widetilde{B_3}\widetilde{B_4}\widetilde{B_5}\widetilde{B_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right] \tag{11}$$

$$\beta = \left[\frac{\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6}{\widetilde{\beta_1}\widetilde{\beta_2}\widetilde{\beta_3}\widetilde{\beta_4}\widetilde{\beta_5}\widetilde{\beta_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right] \tag{12}$$

$$\varepsilon = \left[\frac{\varepsilon_1\varepsilon_2\varepsilon_3\varepsilon_4\varepsilon_5\varepsilon_6}{\widetilde{\varepsilon_1}\widetilde{\varepsilon_2}\widetilde{\varepsilon_3}\widetilde{\varepsilon_4}\widetilde{\varepsilon_5}\widetilde{\varepsilon_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right] \tag{13}$$

Moreover, during computation time the current state of the TM is decoded in the sticky-end of the last cleaved molecule. Let us consider the following DNA strand:

$$N = \left[\frac{N_1N_2N_3N_4N_5N_6}{\widetilde{N_1}\widetilde{N_2}\widetilde{N_3}\widetilde{N_4}\widetilde{N_5}\widetilde{N_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right]$$

1. The following situation reports that the TM is in state S_0 with input N.

$$N = \left[\frac{N_1N_2N_3N_4N_5N_6}{\widetilde{N_5}\widetilde{N_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right]$$

2. The following situation reports that the TM is in state S_1 with input N.

$$N = \left[\frac{N_1N_2N_3N_4N_5N_6}{\widetilde{N_6}}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right]$$

3. The following situation reports that the TM is in state S_2 with input N.

$$N = \left[\frac{N_1N_2N_3N_4N_5N_6}{}\Big|\frac{F_1F_2F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}\right]$$

We further extend our notation in order to make this exposition clearer.

Notation 3. *The string $[X_Y]$ where X is a natural number, and $Y \in \Sigma$, represents X number of nucleotides corresponding to symbol Y. Whereas the string $[X]$ represents X number of arbitrary nucleotides.*

For example, the string $[6_1|4_F]$ is the simplified version of the DNA strand (11) described above. With this in mind we are in position to present the molecular version of the transition rules that were described in the previous section.

$$\begin{gathered} T_1 :< S_0, 0 >\mapsto< S_1, \beta, \Rightarrow> \\ \left[\overrightarrow{BsrDI}|4_F|6_\beta|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|4|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|6|\overleftarrow{BbvI}\right] \end{gathered} \tag{14}$$

$$T_2 :< S_0, 1 >\mapsto< S_2, \beta, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_\beta|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|3|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|6|\overleftarrow{BbvI}\right] \tag{15}$$

$$T_3 :< S_0, \beta >\mapsto< HALT >$$
$$\left[\overrightarrow{BsrDI}|4_F|\ H\ A\ L\ T\ |6_\beta|6|\overleftarrow{BbvI}\right] \tag{16}$$

$$T_4 :< S_1, 0 >\mapsto< S_0, 1, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_1|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|7|\overleftarrow{BbvI}\right] \tag{17}$$

$$T_5 :< S_1, 1 >\mapsto< S_0, 1, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_1|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_1|7|\overleftarrow{BbvI}\right] \tag{18}$$

$$T_6 :< S_1, \beta >\mapsto< S_0, \varepsilon, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_\varepsilon|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_\beta|7|\overleftarrow{BbvI}\right] \tag{19}$$

$$T_7 :< S_2, 0 >\mapsto< S_0, 1, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_1|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|8|\overleftarrow{BbvI}\right] \tag{20}$$

$$T_8 :< S_2, 1 >\mapsto< S_0, 0, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_1|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_0|8|\overleftarrow{BbvI}\right] \tag{21}$$

$$T_9 :< S_2, \beta >\mapsto< S_0, \varepsilon, \Rightarrow >$$
$$\left[\overrightarrow{BsrDI}|4_F|6_\varepsilon|4_F|6|\overleftarrow{BserI}|\overrightarrow{FokI}|5|4_F|12|\overleftarrow{BpmI}|\overrightarrow{BpmI}|8|6_\beta|8|\overleftarrow{BbvI}\right] \tag{22}$$

Where the string

$$[\ H\ A\ L\ T\ ] \tag{23}$$

denotes the double-stranded DNA molecule that acts as detection molecule for the TM, and indicates that the computation is over. The choice of base pairs that comprise this molecule is arbitrary as long as it does not interfere with the action of the restriction enzymes. As done previously, we group these molecules in a mathematical set that we name Γ.

The computation begins when the circular double-stranded DNA molecule, in which we encode the input string, is mixed with all the transition molecules and the restriction enzymes. The *head* of the machine is represented by the recognition sites of restriction enzymes FokI and BserI.

The computation takes place immediately when the restriction enzymes react to the contents of the circular double-stranded DNA molecule, and the transition molecules bind to it via their sticky-ends. At the end of the computation, the contents of the tape can be read on the DNA strand, and this corresponds to the output of the TM.

After having described the contents of the sets Γ (transition molecules), Φ (restriction enzymes) and Σ (alphabet molecules) we are able to announce our main theorem.

Theorem 1. $< \Gamma, \Phi, \Sigma >$ *is a molecular Turing machine that computes a logical NAND function.*

The proof of this theorem can be found in the Appendix section of this paper, which basically proves that our design of transition molecules, alphabet and restriction enzymes achieves to mimic the action of a logical NAND gate on a molecular TM. Thus, showing that it is possible to build a molecular version of a logic unit, which together with an arithmetic unit, is an essential part of a central processing unit of any computing device.

Figure (1) in page 20 shows an schematic example of the TM operation with input `01`, where we show how enzymes cleave the molecule, and how the transition molecules bind to it mimicking the action of the head reading/writing on the tape and then moving along it. For input `01` we expect the machine to return `1` as output (see table (1) above), which is the case for our design.

The action of our TM can be summarised in the following steps, naturally we are assuming ideal conditions, which means that our molecules are not affected by other substances in the medium and computation is carried out flawlessly:

1. Enzymes FokI and BserI recognise their binding sites and cleave the molecule accordingly. Because the molecule is a circular DNA strand, no nucleotide is lost after cleavage.
2. Transition molecules enter the medium after being cleaved by enzyme BbvI.
3. In the medium, enzyme BsrDI acts upon the transition molecules, which makes them susceptible to bind to the main strand.
4. The enzyme ligase aids the coupling of transition molecules and the main strand in virtue of their complementary sticky-ends.
5. To delete the character that has been read from the tape, enzyme BpmI acts upon the DNA strand, which cleaves that particular section of the string.
6. Once more, enzyme ligase bind together the complementary sticky-ends, which results in a complete circular double-stranded DNA molecule.
7. The first step is repeated until no recognition site is found within the main molecule, which implies that the string representing the *HALT* state is contained in the molecule.

3 Discussion

An important factor that must be taken in consideration when implementing this kind of molecular devices in the lab is the sequence of steps in which computation

takes place. In our mathematical abstraction, these steps occur one at a time and without errors on a unique DNA molecule. In reality, we would have thousands of these molecules, plus restriction enzymes and other molecules required for the computation (e.g. ligase and ATP) in a single test tube, all of them influencing each other in one way or the other. Therefore, we would expect that only a percentage of those strands will perform the computation as planned.

Should this fact discourage us? We should keep in mind that the molecular computing paradigm is still in its infancy, and a lot of work must be done both in the theoretical and in the experimental fronts. Research efforts must be directed towards developing mechanisms for error detection and fault tolerance. In [**?**], the author presents a series of strategies to deal with undesired effects during molecular runtime. Examples of these adverse effects are: faulty or incorrect bonds, and faulty or incorrect cuts. Strategies to deal with them include the use of exonucleases and endonucleases with no specific restriction site to detect and correct errors during run-time [**?**].

Genetic material is a promising computing medium due to its relative ease of parallelisation and its natural tendency to bind to complementary strings. This features offer the possibility of solving hard computing problems with less effort and cost than it would require to conventional computation. The cost of building and maintaing a computer cluster is very high in terms of space, energy and other human and technical resources. On the other hand, molecular computing offers the possibility of having thousands of molecules computing in a space as small as a teaspoon, without any negative environmental impact.

Nevertheless, it is very likely that up to now this computer paradigm might not perform any better than our modern computers. However, we should take a few steps back and think about the nature of this new computing paradigm. Genetic material as a computing device opens a new perspective on the kind of problems that we could face with it. DNA is the native language of the cell, therefore using the former as software and the latter as hardware gives us a new opportunity to deal with medical conditions that remain intractable to date. One of the first steps towards this direction is the construction of an arithmetic and logic unit made entirely from DNA molecules. Here we offered an insight for such a purpose.

Appendix

Theorem 1. $< \Gamma, \Phi, \Sigma >$ *is a molecular Turing machine that computes a logical NAND function.*

Proof. We proceed in the following way. Let us consider a string as long as the proof requires, that is, a circular double-stranded DNA molecule with undefined length. Thus, we have the following initial configuration:

$$(6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|9|6_{\lambda_1}|4_F|6_{\lambda_2}|4_F|\cdots|6_{\lambda_i}|4_F|\cdots) \tag{24}$$

At this point, the head of the machine is *placed over* the character λ_1. The next step occurs when enzymes FokI and BserI recognise their sites and cleave

the molecule. This action leaves the molecule in the following configuration:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\ |\frac{\lambda_{11}\lambda_{12}\lambda_{13}\lambda_{14}\lambda_{15}\lambda_{16}}{\widetilde{\lambda_{15}}\widetilde{\lambda_{16}}}|4_F|6_{\lambda_2}|4_F|\cdots\right) \tag{25}$$

The sticky-end $\lambda_{11}\lambda_{12}\lambda_{13}\lambda_{14}$ indicates that the machine is currently in state S_0 with input λ_1. Because the molecule is circular, that is, it is linked by its extremes, the computation will proceed and a molecule has a chance to fill the gap caused by the action of the enzymes, provided it has a complementary sticky-end. Let us analyse the different cases for the value of character λ_1.

CASE 1: $< S_0, \lambda_1 = \beta >$ Then, string (25) is actually the string:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\ |\frac{\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6}{\widetilde{\beta_5}\widetilde{\beta_6}}|4_F|6_{\lambda_2}|4_F|\cdots\right) \tag{26}$$

And thus, we expect transition T_3 (16) to take place. The molecule representing such transition gets close to the empty space in string (26) as shown below:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\qquad\qquad\qquad\qquad\qquad\qquad|\frac{\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6}{\widetilde{\beta_5}\widetilde{\beta_6}}|4_F|\cdots\right) \tag{27}$$
$$|\overrightarrow{\overline{BsrDI}}|4_F|\ H\ A\ L\ T\ |6_\beta|6|\overleftarrow{\overline{BbvI}}|$$

However, this molecule cannot incorporate to the main strand yet, as it does not exhibit any sticky-end. This happens after enzymes BsrDI and BbvI act on it, which results in the following string:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\qquad\qquad\qquad\qquad\qquad\qquad|\frac{\beta_1\beta_2\beta_3\beta_4\beta_5\beta_6}{\widetilde{\beta_5}\widetilde{\beta_6}}|4_F|\cdots\right) \tag{28}$$
$$|\overline{\overline{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}^{F_3F_4}}|\ H\ A\ L\ T\ |\overline{\overline{\widetilde{\beta_1}\widetilde{\beta_2}\widetilde{\beta_3}\widetilde{\beta_4}}}|$$

As the sticky-ends in both molecules are complementary, they are susceptible of binding together, which results in the following molecule:

$$(6_\beta|4_F|\ H\ A\ L\ T\ |6_\beta|4_F|\cdots) \tag{29}$$

Resulting in a complete molecule with no recognition sites on which an enzyme could act. Thus, at this stage the computation ends as expected.

CASE 2: $< S_0, \lambda_1 = 0 >$ Thus, string (25) is actually the string:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\ |\frac{A_1A_2A_3A_4A_5A_6}{\widetilde{A_5}\widetilde{A_6}}|4_F|6_{\lambda_2}|4_F|\cdots\right) \tag{30}$$

In this case, transition T_1 (14) should take place, which at the moment of attempting to bind to the main molecule results in:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}|\qquad\qquad\qquad\qquad\qquad\qquad|\frac{A_1A_2A_3A_4A_5A_6}{\widetilde{A_5}\widetilde{A_6}}|\cdots\right) \tag{31}$$
$$|\overrightarrow{\overline{BsrDI}}|4_F|6_\beta|4_F|6|\overleftarrow{\overline{BserI}}|\overrightarrow{\overline{FokI}}|4|4_F|12|\overleftarrow{\overline{BpmI}}|\overrightarrow{\overline{BpmI}}|8|6_0|6|\overleftarrow{\overline{BbvI}}|$$

After enzymes BsrDI and BbvI act upon transition molecule T_1 they yield sticky-ends in both extremes which are susceptible of binding to the main molecule, which results in:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}| \qquad\qquad |\overline{\overline{}^{A_1A_2A_3A_4A_5A_6}}|\cdots\right) \\ |\overline{\overline{}^{F_3F_4}}|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BserI}}|\overrightarrow{\overrightarrow{FokI}}|4|4_F|12|\overleftarrow{\overleftarrow{BpmI}}|\overrightarrow{\overrightarrow{BpmI}}|8|\overline{\overline{}}| \tag{32}$$

And then,

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BserI}}|\overrightarrow{\overrightarrow{FokI}}|4|4_F|12|\overleftarrow{\overleftarrow{BpmI}}|\overrightarrow{\overrightarrow{BpmI}}|8|6_0|4_F|6_{\lambda_2}|4_F|\cdots) \tag{33}$$

In the next step, the enzyme BpmI recognises its site and binds to the molecule cleaving it, which results in:

$$\left(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BseRI}}|\overrightarrow{\overrightarrow{FokI}}|4|\overline{\overline{}^{F_1F_2}}| \ |\overline{\overline{}^{F_3F_4}}|6_{\lambda_2}|4_F|\cdots\right) \tag{34}$$

where the sticky-ends are susceptible of binding, which yields the following configuration:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BseRI}}|\overrightarrow{\overrightarrow{FokI}}|4|4_F|6_{\lambda_2}|4_F|\cdots) \tag{35}$$

which we consider as a new initial configuration for the machine, as we did for the configuration (24). At this point, the head is *placed over* the character λ_2. We will continue exploring this branch of the computation later in this proof (see below). We continue developing the different cases of λ_1.

CASE 3: $< S_0, \lambda_1 = 1 >$ So, the string (25) is actually the following string:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}| \ |\overline{\overline{}^{B_1B_2B_3B_4B_5B_6}}|4_F|6_{\lambda_2}|4_F|\cdots\right) \tag{36}$$

At this moment, transition T_2 should take place. Such a molecule (15) attempts to bind to the main molecule which results in the following:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}| \qquad\qquad |\overline{\overline{}^{B_1B_2B_3B_4B_5B_6}}|\cdots\right) \\ |\overrightarrow{\overrightarrow{BsrDI}}|4_F|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BseRI}}|\overrightarrow{\overrightarrow{FokI}}|3|4_F|12|\overleftarrow{\overleftarrow{BpmI}}|\overrightarrow{\overrightarrow{BpmI}}|8|6_1|6|\overleftarrow{\overleftarrow{BbvI}}| \tag{37}$$

After enzymes BsrDI and BbvI act upon this transition molecule, we obtain the following pair of strings:

$$\left(6_\beta|\overline{\overline{}^{F_1F_2}}| \qquad\qquad |\overline{\overline{}^{B_1B_2B_3B_4B_5B_6}}|\cdots\right) \\ |\overline{\overline{}^{F_3F_4}}|6_\beta|4_F|6|\overleftarrow{\overleftarrow{BseRI}}|\overrightarrow{\overrightarrow{FokI}}|3|4_F|12|\overleftarrow{\overleftarrow{BpmI}}|\overrightarrow{\overrightarrow{BpmI}}|8|\overline{\overline{}}| \tag{38}$$

The sticky-ends of the last molecule are susceptible of binding together, which results in the following molecule:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|3|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_1|4_F|6_{\lambda_2}|4_F|\cdots) \tag{39}$$

Next, enzyme BpmI recognises its binding site and cleaves the molecule resulting in:

$$\left(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|3|\overline{^{F_1F_2}}|\ |\underset{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}{\overline{^{F_3F_4}}}|6_{\lambda_2}|4_F|\cdots\right) \tag{40}$$

The sticky-ends of the last molecule are susceptible of binding together, which yields the following configuration:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|3|4_F|6_{\lambda_2}|4_F|\cdots) \tag{41}$$

which analogously to **CASE 2** is a new configuration for the machine, whose development will be studied below. At this moment, the head is reading character λ_2 after writing character β as expected.

With our previous exposition, we have exhausted all the possible cases for character λ_1, we proceed now to explore the branches of the computation that resulted from developing the first character. Given that **CASE 1** represents a state in which the machine has stopped, we analyse the other two remaining case and branch accordingly. Now, we explore the **CASE 2** branch, that is, when $\lambda_1 = 0$, we have as initial configuration the strand:

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|4|4_F|6_{\lambda_2}|4_F|\cdots) \tag{42}$$

The head is currently over the character λ_2. When enzymes FokI and BserI act upon this string, they yield the following string:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\underset{\widetilde{\lambda_{26}}}{\overline{^{\lambda_{22}\lambda_{23}\lambda_{24}\lambda_{25}\lambda_{26}}}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{43}$$

where the sticky-end $\lambda_{22}\lambda_{23}\lambda_{24}\lambda_{25}$ reports that the machine is currently in state S_1 with input λ_2, which was expected from the transition rules after reading character 0 on the tape. Let us explore the different cases for λ_2.

CASE 2.1: $< S_1, \lambda_2 = \beta >$ So, the string (43) is actually the string:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\underset{\widetilde{\beta_6}}{\overline{^{\beta_2\beta_3\beta_4\beta_5\beta_6}}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{44}$$

According to our transition rules, when the machine reads character β in state S_1, transition T_6 takes place that is, molecule (19) attempts to bind to the main molecule, which looks like:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\underset{\widetilde{\beta_6}}{\overline{^{\beta_2\beta_3\beta_4\beta_5\beta_6}}}|\cdots\right)$$
$$|\overrightarrow{_{BsrDI}}|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_\beta|7|\overleftarrow{_{BbvI}}| \tag{45}$$

Once the enzymes BsrDI and BbvI have cleaved the transition molecule, the string becomes:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}| \qquad\qquad |\frac{\beta_2\beta_3\beta_4\beta_5\beta_6}{\widetilde{\beta_6}}|\cdots\right)$$
$$|\frac{F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\frac{\beta_1}{\widetilde{\beta_1}\widetilde{\beta_2}\widetilde{\beta_3}\widetilde{\beta_4}\widetilde{\beta_5}}| \tag{46}$$

Here, the sticky-ends are susceptible of binding together. When this happens, the string becomes:

$$(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_\beta|4_F|6_{\lambda_3}|4_F|\cdots) \tag{47}$$

Then, enzyme BpmI cleaves the molecule, which results in the following:

$$\left(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{}^{F_1F_2}|\ |\frac{F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_{\lambda_3}|4_F|\cdots\right) \tag{48}$$

We see sticky-ends that are susceptible of binding together. When this happens the resulting molecule is:

$$(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \tag{49}$$

Up to this point, we know that the head is placed over the character λ_3 and it has written the symbol ε on the tape. This string can be considered as a new initial configuration for the machine, but before we continue exploring its branch, we take a look at the remaining cases.

CASE 2.2: $< S_1, \lambda_2 = 0 >$ So, the string (43) is actually the string:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}|\ |\frac{A_2A_3A_4A_5A_6}{\widetilde{A_6}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{50}$$

Given the current state of the machine and the character being read, the transition T_4 must take place, and thus molecule (17) attempts to add to the main molecule:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}| \qquad\qquad |\frac{A_2A_3A_4A_5A_6}{\widetilde{A_6}}|\cdots\right)$$
$$|\overrightarrow{_{BsrDI}}|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_0|7|\overleftarrow{_{BbvI}}| \tag{51}$$

When enzymes BsrDI and BbvI act upon the transition molecule they produce the following string:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}| \qquad\qquad |\frac{A_2A_3A_4A_5A_6}{\widetilde{A_6}}|\cdots\right)$$
$$|\frac{F_3F_4}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\frac{A_1}{\widetilde{A_1}\widetilde{A_2}\widetilde{A_3}\widetilde{A_4}\widetilde{A_5}}| \tag{52}$$

which has sticky-ends susceptible of binding together. When they do so, the resulting molecule is:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_0|4_F|6_{\lambda_3}|4_F|\cdots) \quad (53)$$

Next, the enzyme BpmI cleaves the molecule, which results in:

$$\left(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{^{F_1F_2}}|\ |\frac{^{F_3F_4}}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_{\lambda_3}|4_F|\cdots\right) \quad (54)$$

Again, the molecule has sticky-ends susceptible of binding together. Thus, the molecule becomes:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \quad (55)$$

At this point, the head is placed over the character λ_3 after writing character 1 on the tape., which is expected after reading characters 00 on the tape. The new state is S_0 and below we describe what happens if we further elaborate this case.

CASE 2.3: $< S_1, \lambda_2 = 1 >$ Then, string (43) is actually the string:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\frac{^{B_2B_3B_4B_5B_6}}{\widetilde{B_6}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \quad (56)$$

In this situation, the machine is in state S_1 and input $\lambda_2 = 1$, thus transition T_5 should take place. When the molecule representing this transition (18) attempts to bind to the main molecule, we obtain:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\frac{^{B_2B_3B_4B_5B_6}}{\widetilde{B_6}}|\cdots\right) |\overrightarrow{_{BsrDI}}|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_1|7|\overleftarrow{_{BbvI}}| \quad (57)$$

Here, enzymes BsrDI and BbvI recognise their sites and cleave the molecule, which results in:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\frac{^{B_2B_3B_4B_5B_6}}{\widetilde{B_6}}|\cdots\right) |\frac{^{F_3F_4}}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\frac{^{B_1}}{\widetilde{B_1}\widetilde{B_2}\widetilde{B_3}\widetilde{B_4}\widetilde{B_5}}| \quad (58)$$

This allows the sticky-ends to bind together, which produces the following strand:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_1|4_F|6_{\lambda_3}|4_F|\cdots) \quad (59)$$

Finally, the enzyme BpmI cleaves the molecules producing:

$$\left(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{^{F_1F_2}}|\ |\frac{^{F_3F_4}}{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}|6_{\lambda_3}|4_F|\cdots\right) \quad (60)$$

And after the sticky-ends bind together, we obtain the molecule:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \tag{61}$$

Which can be considered a new initial configuration for the machine. The head is now placed over the character λ_3 in state S_0 after writing character 1 on the tape, as expected after reading the string 01.

We wonder what would happen if we further developed the recently described cases. We answer this question in the following observation.

Observation 1. *Let us take a look at the tapes that result from following the three last cases (**2.1,2.2,2.3**). First thing we observe is that strings (55) and (61) are the same. Moreover, strings (49), (55) y (61) correspond to state S_0 with input λ_3 differing only in the contents of its tape. This strings are similar to the initial configuration of the machine, string (24), and thus, we expect to get the same results when branching from it, no matter the current contents of the tape. In other words, we would expect the same molecular reactions to occur, which has been described above. This is valid when considering $\lambda_1 = 0$. On the other hand, if we consider $\lambda_1 = 1$ we can further extend the computing branch of the machine. We proceed in this direction below.*

We have the following DNA strand as initial configuration for the machine, in which the head is reading character λ_2.

$$(6_\beta|4_F|6_\beta|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|3|4_F|6_{\lambda_2}|4_F|\cdots) \tag{62}$$

Enzymes FokI and BserI act upon this molecule, so that we obtain the following:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\overline{^{\lambda_{23}\lambda_{24}\lambda_{25}\lambda_{26}}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{63}$$

Here, the sticky-end reports that the machine is currently in state S_2 with input λ_2. Once again, we explore the different cases for the value of λ_2.

CASE 3.1: $< S_2, \lambda_2 = \beta >$ So, the string (63) is actually the string:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\overline{^{\beta_3\beta_4\beta_5\beta_6}}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{64}$$

Given the input $\lambda_2 = \beta$ and state S_2 we expect transition T_9 to take place. This transition is given by the molecule (22) and on attempting to bind to the main molecule results in:

$$\left(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\overline{^{\beta_3\beta_4\beta_5\beta_6}}|\cdots\right)$$
$$|\overrightarrow{_{BsrDI}}|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_\beta|8|\overleftarrow{_{BbvI}}| \tag{65}$$

After enzymes BsrDI and BbvI have recognised their sites, the molecule becomes:

$$\Big(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\overline{^{\beta_3\beta_4\beta_5\beta_6}}|\cdots\Big)$$
$$|\overline{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}^{F_3F_4}|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\overline{\widetilde{\beta_1}\widetilde{\beta_2}\widetilde{\beta_3}\widetilde{\beta_4}\widetilde{\beta_5}\widetilde{\beta_6}}^{\beta_1\beta_2}| \tag{66}$$

Which binds to the main molecule producing:

$$(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_\beta|4_F|6_{\lambda_3}|4_F|\cdots) \tag{67}$$

Next, enzyme BpmI recognises its site and cleaves the molecule, thus producing:

$$\Big(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{^{F_1F_2}}|\ |\overline{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}^{F_3F_4}|6_{\lambda_3}|4_F|\cdots\Big) \tag{68}$$

When the complementary sticky-ends of this molecule bind together the molecule becomes:

$$(6_\beta|4_F|6_\beta|4_F|6_\varepsilon|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \tag{69}$$

Which is a new initial configuration for the machine, and will be analysed later in this proof, meanwhile we know that the head is placed over character λ_3 and it has just written character ε on the tape as it was expected from input 1β.

CASE 3.2: $< S_2, \lambda_2 = 0 >$ So, string (63) is actually the string:

$$\Big(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}|\ |\overline{^{A_3A_4A_5A_6}}|4_F|6_{\lambda_3}|4_F|\cdots\Big) \tag{70}$$

For input $\lambda_2 = 0$ in state S_2 we expect transition T_7 to occur. Thus, the molecule becomes:

$$\Big(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\overline{^{A_3A_4A_5A_6}}|\cdots\Big)$$
$$|\overrightarrow{_{BsrDI}}|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_0|8|\overleftarrow{_{BbvI}}| \tag{71}$$

After enzymes BsrDI and BbvI have cleaved the molecule, we obtain:

$$\Big(6_\beta|4_F|6_\beta|\overline{^{F_1F_2}}| \qquad\qquad |\overline{^{A_3A_4A_5A_6}}|\cdots\Big)$$
$$|\overline{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}^{F_3F_4}|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\overline{\widetilde{A_1}\widetilde{A_2}\widetilde{A_3}\widetilde{A_4}\widetilde{A_5}\widetilde{A_6}}^{A_1A_2}| \tag{72}$$

Here, the sticky-ends are complementary, which means that they are susceptible of binding together. After this happens we obtain:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_0|4_F|6_{\lambda_3}|4_F|\cdots) \tag{73}$$

Next, enzyme BpmI acts upon the string, which becomes:

$$\left(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{}^{F_1F_2}|\ |\underset{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}{\overline{}^{F_3F_4}}|6_{\lambda_3}|4_F|\cdots\right) \tag{74}$$

Which becomes the following strand after the sticky-ends bind together:

$$(6_\beta|4_F|6_\beta|4_F|6_1|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \tag{75}$$

This last strand can be considered as a new initial configuration for the machine. We develop this branch of the computation below. Meanwhile, it is enough to say that currently the head is placed over character λ_3, and it has just written the character 1 on the tape.

CASO 3.3: $< S_2, \lambda_2 = 1 >$ Then, the string (63) is actually the string:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}|\ |\overline{}^{B_3B_4B_5B_6}|4_F|6_{\lambda_3}|4_F|\cdots\right) \tag{76}$$

For this combination of input and state we expect transition T_8. When the molecule (21) representing this transition attempts to bind to the main molecule, we see:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}|\qquad\qquad\qquad\qquad|\overline{}^{B_3B_4B_5B_6}|\cdots\right)$$
$$|\overrightarrow{_{BsrDI}}|4_F|6_0|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_1|8|\overleftarrow{_{BbvI}}| \tag{77}$$

This molecule is not yet able to bind to the main molecule. It is only when enzymes BsrDI and BbvI recognise their sites that the transition molecule exhibits sticky-ends that are complementary to the main molecule:

$$\left(6_\beta|4_F|6_\beta|\overline{}^{F_1F_2}|\qquad\qquad\qquad\qquad|\overline{}^{B_3B_4B_5B_6}|\cdots\right)$$
$$|\underset{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}{\overline{}^{F_3F_4}}|6_0|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|\underset{\widetilde{B_1}\widetilde{B_2}\widetilde{B_3}\widetilde{B_4}\widetilde{B_5}\widetilde{B_6}}{\overline{}^{B_1B_2}}| \tag{78}$$

When the complementary sticky-ends bing together, they yield the following molecule:

$$(6_\beta|4_F|6_\beta|4_F|6_0|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|12|\overleftarrow{_{BpmI}}|\overrightarrow{_{BpmI}}|8|6_1|4_F|6_{\lambda_3}|4_F|\cdots) \tag{79}$$

Next, the enzyme BpmI cleaves the molecule after recognising its site. This results in:

$$\left(6_\beta|4_F|6_\beta|4_F|6_0|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|\overline{}^{F_1F_2}|\ |\underset{\widetilde{F_1}\widetilde{F_2}\widetilde{F_3}\widetilde{F_4}}{\overline{}^{F_3F_4}}|6_{\lambda_3}|4_F|\cdots\right) \tag{80}$$

After the complementary sticky-ends bind together, we obtain the following molecule:

$$(6_\beta|4_F|6_\beta|4_F|6_0|4_F|6|\overleftarrow{_{BseRI}}|\overrightarrow{_{FokI}}|5|4_F|6_{\lambda_3}|4_F|\cdots) \tag{81}$$

This strand can be considered a new initial configuration. Moreover, we know that the head is currently placed over the character λ_3 and that the character 0 has just been written recently on the tape.

At this point, it is natural to wonder what happens if we further develop the **CASES** recently presented. In the following observation we give an answer to this question.

Observation 2. *Let us take a look at the tapes that result from applying* ***CASES 3.1, 3.2 y 3.3****, that is, strings (69), (75) and (81). As noted previously, these strings are similar to each other and to the initial configuration (24). Thus, we can expect the same behaviour from our TM for the rest of the computation. In other words, we have exhausted all the possible behaviours of the machine, thus its operation will ever fall in one of the cases described above.*

The above not only proves that our transition molecules work correctly on the molecular tape, but also that they write what is expected on the tape. □

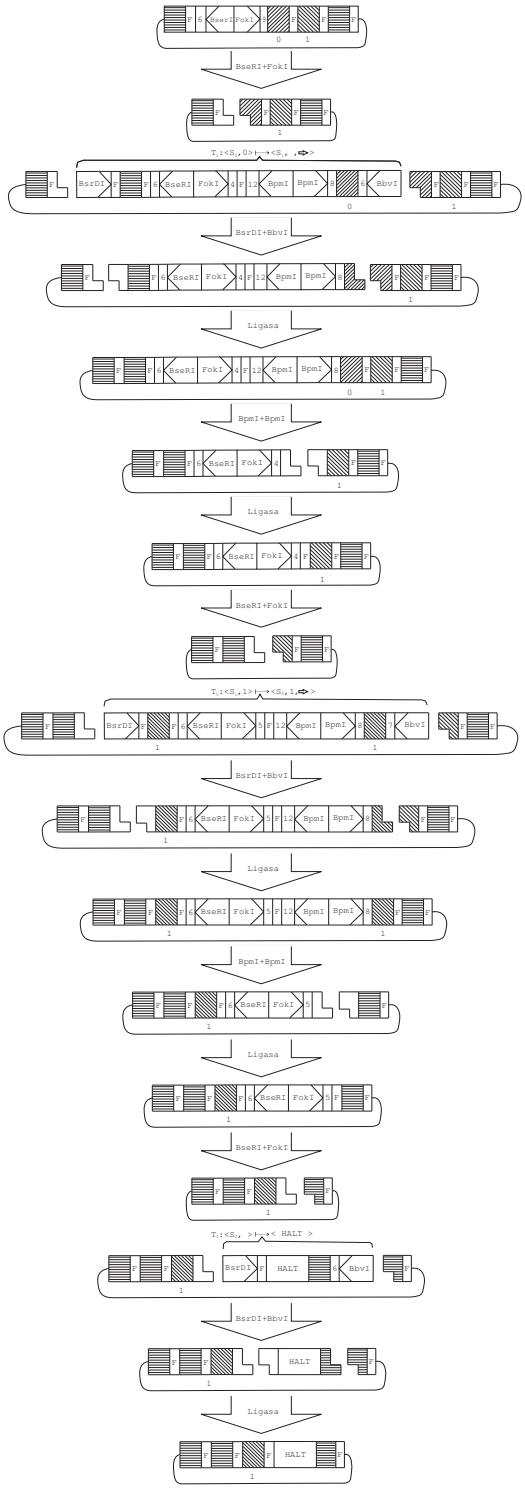

Fig. 1. Molecular TM working on input `01`